Appendix

A Axioms of the System

There are three sets of axioms.

Specific axioms for features and sorts :
Let τ, τ' denote any sorts, and f denote any feature.

$$\begin{array}{ll}
\forall x, y, z & x \xrightarrow{f} y \wedge x \xrightarrow{f} z \supset y = z \\
\forall x & \neg(x : \tau \wedge x : \tau') \text{ if } \tau \neq \tau' \\
\forall x, y & x : \tau \wedge y : \tau \supset x = y \text{ if } \tau \text{ is a value sort} \\
\forall x, y & \neg(x : \tau \wedge x \xrightarrow{f} y) \text{ if } \tau \text{ is a value sort}
\end{array}$$

Congruence axioms for equality :
Let p denote any built-in predicate. The traditional congruence axioms are:

$$\begin{array}{ll}
\forall x & x = x \\
\forall x, y & x = y \supset y = x \\
\forall x, y, z & x = y \wedge y = z \supset x = z \\
\forall x, y & x : \tau \wedge x = y \supset y : \tau \\
\forall x, y, z & x \xrightarrow{f} y \wedge x = z \supset z \xrightarrow{f} y \\
\forall x, y, z & x \xrightarrow{f} y \wedge y = z \supset x \xrightarrow{f} z \\
\forall \bar{x}, y & p(\bar{x}) \wedge x_i = y \supset p(\bar{y})
\end{array}$$

where i is some index in the list of variable $\bar{x}$ and $\bar{y}$ is identical to $\bar{x}$ except that $y_i = y$.

Built-in predicate axioms :
They must not mention sorts and features. For example, disequality can be axiomatized by

$$\begin{array}{ll}
\forall x, y & x \neq y \vee x = y \\
\forall x & \neg(x \neq x)
\end{array}$$

Precedence constraints are axiomatized by

$$\begin{array}{ll}
\forall x & \neg(x < x) \\
\forall x, y, z & x < y \wedge y < z \supset x < z
\end{array}$$

The built-in predicates $>, \leq, \geq$ can then be defined from $<$ and equality.

B Constraint Satisfaction

B.1 The BFC case

We represent a BFC as a pair $\langle B \mid \Gamma \rangle$ where B is a built-in constraint and Γ an *unordered* list of sort and feature constraints (read conjunctively). $\perp$ denotes the contradiction.

There are two sets of rewrite rules. The following rules correspond to simplifications of the BFCs.

$$
\begin{array}{lcl}
\langle B \mid x \xrightarrow{f} y, z \xrightarrow{f} t, \Gamma\rangle & \longmapsto & \langle B \wedge y = t \mid x \xrightarrow{f} y, \Gamma\rangle \text{ if } \vdash_{\mathcal{T}} B \supset x = z \\
\langle B \mid x : \tau, y : \tau, \Gamma\rangle & \longmapsto & \langle B \mid x : \tau, \Gamma\rangle \text{ if } \vdash_{\mathcal{T}} B \supset x = y \text{ and } \tau \text{ is not a value sort} \\
\langle B \mid x : \tau, y : \tau, \Gamma\rangle & \longmapsto & \langle B \wedge x = y \mid x : \tau, \Gamma\rangle \text{ if } \tau \text{ is a value sort}
\end{array}
$$

The following rules correspond to the detection of inconsistencies.

$$
\begin{array}{lcl}
\langle B \mid \Gamma\rangle & \longmapsto & \bot \text{ if } \vdash_{\mathcal{T}} \neg B \\
\langle B \mid x : \tau, y : \tau', \Gamma\rangle & \longmapsto & \bot \text{ if } \vdash_{\mathcal{T}} B \supset x = y \text{ and } \tau \neq \tau' \\
\langle B \mid x : \tau, y \xrightarrow{f} z, \Gamma\rangle & \longmapsto & \bot \text{ if } \vdash_{\mathcal{T}} B \supset x = y \text{ and } \tau \text{ is a value sort}
\end{array}
$$

The following property justifies the algorithm

$$\langle B \mid \Gamma\rangle \longmapsto \bot \text{ if and only if } \vdash_{\mathcal{T}} \forall \neg (B \wedge \bigwedge_{c \in \Gamma} c)$$

B.2 The SFC case

We represent an SFC as an *unordered* list of BFCs prefixed with a sign (+ or −); by definition, one and only one component is positive. Let $\mathcal{S}$ be an SFC. The SFC-normal form of $\mathcal{S}$ is written $\mathcal{S}^*$ and is obtained by the following algorithm:

Let c_o be the BFC-normal form of the positive component of $\mathcal{S}$.
If $c_o = \bot$ **Then**
  Return $\bot$
Else
  c_o is of the form $\langle B_o \mid \Gamma_o\rangle$
  Let $\{\langle B_i \mid \Gamma_i\rangle\}_{i=1,\ldots,n}$ be the list of negative components of $\mathcal{S}$.
  Foreach $i = 1, \ldots, n$
    Let c_i be the BFC normal form of $\langle B_o \wedge B_i \mid \Gamma_o, \Gamma_i\rangle$.
  If there exists $i \in 1, \ldots, n$ such that $c_i = \langle B \mid \Gamma\rangle$ and $\vdash_{\mathcal{T}} B$ and Γ is empty **Then**
    Return $\bot$
  Else
    Let $I = \{i \in 1, \ldots, n \text{ such that } c_i \neq \bot\}$
    Return $\{+c_o, \{-c_i\}_{i \in I}\}$

The following property justifies the algorithm

$$[+\langle B_o \mid \Gamma_o\rangle, \{-\langle B_i \mid \Gamma_i\rangle\}_{i=1}^{n}]^* = \bot \text{ if and only if } \vdash_{\mathcal{T}} \forall \neg [(B_o \wedge \bigwedge_{c \in \Gamma_o} c) \wedge \bigwedge_{i=1}^{n} \neg (B_i \wedge \bigwedge_{c \in \Gamma_i} c)]$$